UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/839,526	04/20/2001	Dietrich Charisius	30013630-0009	2147

23485 7590 06/04/2004

JINAN GLASGOW
P O BOX 28539
RALEIGH, NC 276118539

EXAMINER	
SHRADER, LAWRENCE J	

ART UNIT	PAPER NUMBER
2124	

DATE MAILED: 06/04/2004

11

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/839,526	CHARISIUS ET AL.
	Examiner	Art Unit
	Lawrence Shrader	2124

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 4/20/2001,7/16/2001,8/10/2001,8/15/200.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-136 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-136 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Information Disclosure Statement

1. The application file wrapper indicates that the Office received two Information Disclosure Statements (IDS) with the application on the 7/16/2001, 8/10/2004, and 8/15/2004. However, the Examiner cannot find either IDS in the application. Therefore, the IDSs were not considered because they are absent from the file folder.

2. Applicant is requested to update the status of the applications referenced in the Cross Reference section of the specification beginning on page 1.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 1, 2, 9 – 11, and 13 – 17; 18, 20, 21, 23; 62, 62, 70 – 72, and 74 – 78; 79, 81, 82, and 84; and 123 - 126 are rejected under 35 U.S.C. 102(e) as being anticipated by Hicks, U.S. Patent 6,654,954.

In regard to claim 1:

1. A method in a data processing system, comprising the steps of:

> *"receiving an identification of a data structure with an attribute field in a database;*
>
> *determining whether the data structure is associated with source code;*
>
> *when it is determined that the data structure is associated with source code, determining whether the attribute field of the data structure is associated with an attribute in the source code; and*
>
> *when it is determined that the attribute field is not associated with an attribute in the source code, generating a new attribute in the source code from the attribute field."*

Hicks discloses the identification of a data structure with an attribute field (file header attributes), determines whether the data structure is associated with source code (file attribute associated with alternate code). Determines whether an attribute field in the data structure is associated with an attribute in the source code (time stamp or checksum information). If the source code associated with the attribute does not exist, the code is generated with associated attributes (column 11, lines 6 – 21; column 10, lines 1 – 10; e.g., Figure 3).

In regard to claim 2, incorporating the rejection of claim 1:

> *"...further comprising the steps of:*
> *when it is determined that the data structure is not associated with source code, retrieving a portion of the data structure; and*
> *generating the source code from the portion of the data structure."*

See Hicks column 11, lines 6 – 21.

In regard to claim 9, incorporating the rejection of claim 1:

> *"...wherein the step of determining whether the data structure is associated with the source code comprises the step of comparing a name for the source code with the identification of the data structure.*

Hicks discloses a filename as an entry in the data structure (column 9, lines 60 – 63; e.g., Figure 3).

In regard to claim 10, incorporating the rejection of claim 1:

> *"...further comprising the steps of:*
> *retrieving access information for the database; and*
> *retrieving a portion of the data structure from the database using the access information."*

Hicks discloses access information in the data structures, see Figure 3.

In regard to claims 11 and 13, incorporating the rejection of claim 10:

> *"...wherein the step of retrieving the access information comprises the step of retrieving the identification of the data structure and the access information from a configuration file;*
> *...wherein the portion of the data structure comprises the attribute field of the data structure."*

Both are rejected for the same reason put forth in the rejection of claim 10.

In regard to claim 14, incorporating the rejection of claim 1:

> *"...wherein the source code comprises a class."*

Hicks discloses the source code as a class (column 15, lines 10 – 12; e.g., Figure 8).

In regard to claim 15, incorporating the rejection of claim 1:

> *"...wherein the source code comprises a distributed computing component."*

Hicks discloses a distributed computing environment (e.g., Figure 1).

In regard to claim 16, incorporating the rejection of claim 15:

"...wherein the distributed computing component is an Enterprise JavaBean™."

Hicks discloses implementation of a distributed system using JAVA. The JAVA language inherently contains appropriate declarations wherein the JAVA classes can be implemented as Javabeans.

In regard to claim 17, incorporating the rejection of claim 1:

"...wherein the step of generating the new attribute in the source code comprises the step of generating a method in the source code to access the attribute field of the data structure."

Hicks discloses the identification of a data structure with an attribute field (file header attributes), determines whether the data structure is associated with source code (file attribute associated with alternate code). Determines whether an attribute field in the data structure is associated with an attribute in the source code (time stamp or checksum information). If the source code associated with the attribute does not exist, the code is generated with associated attributes (column 11, lines 6 – 21; column 10, lines 1 – 10; e.g., Figure 3).

In regard to claim 18:

"receiving an indication to update the source code:

determining whether a first attribute in the source code is associated with a first attribute field in the data structure;

when it is determined that a first attribute in the source code is not associated with a first attribute field in the data structure, removing the first attribute from the source code;

determining whether a second attribute field in the data structure is associated with a second attribute in the source code; and

when it is determined that a second attribute field is not associated with a second attribute in the source code, generating the second attribute in the source code from the second attribute field and associating the second attribute with the second attribute field."

Hicks discloses the identification of a data structure with an attribute field (file header attributes), determines whether the data structure is associated with source code (file attribute associated with alternate code). Determines whether an attribute field in the data structure is associated with an attribute in the source code (time stamp or checksum information). If the source code associated with the attribute does not exist, the code is generated with associated attributes (column 11, lines 6 – 21; column 10, lines 1 – 10; e.g., Figure 3).

In regard to claim 20, incorporating the rejection of claim 19:

"...further comprising the step of retrieving a portion of the data structure from the secondary storage device using the access information."

Hicks discloses access information in the data structures, see Figure 3.

In regard to claim 21, incorporating the rejection of claim 20:

"...wherein the portion comprises the first and the second attribute fields."

Hicks discloses multiple attribute fields (e.g., Figure 3).

In regard to claim 23, incorporating the rejection of claim 18:

"...wherein the step of generating the second attribute in the source code comprises the step of generating a method in the source code to access the second attribute field of the data structure."

Hicks discloses the identification of a data structure with an attribute field (file header attributes), determines whether the data structure is associated with source code (file attribute associated with alternate code). Determines whether an attribute field in the data structure is associated with an attribute in the source code (time stamp or checksum information). If the source code associated with the attribute does not exist, the code is generated with associated attributes (column 11, lines 6 – 21; column 10, lines 1 – 10; e.g., Figure 3).

Claims 62, 62, 70 – 72, and 74 – 78 (computer readable medium) are rejected for the same corresponding reasons put forth in the rejection of claims 1, 2, 9 – 11, and 13 – 17 (the corresponding methods).

Claims 79, 81, 82, and 84 (computer readable medium) are rejected for the same corresponding reasons put forth in the rejection of claims 18, 20, 21 and 23 (the corresponding methods).

Claims 123 - 126 (a data processing system) are rejected for the same corresponding reasons put forth in the rejection of claims 18 – 23 (the corresponding methods).

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 3 – 4; 22; 24 – 33; 34 – 44; 45 – 51; 52 – 61; 64, 65; 83; 85 – 94; 95 – 105; 106 – 112; 113 – 122, 129 – 135; and 136; are rejected under 35 U.S.C. 103(a) as being unpatentable over Hicks, U.S. Patent 6,654,954 in view of Timbol, U.S. Patent 6,237,135.

In regard to claim 3, incorporating the rejection of claim 1:

> *"...further comprising the steps of:*
> *when it is determined that the data structure is associated with source code,*
>
> *determining whether a second attribute in the source code is associated with a second attribute field in the data structure; and*
>
> *when it is determined that a second attribute in the source code is not associated with a second attribute field in the data structure, removing the second attribute from the source code."*

Hicks discloses the identification of a data structure with an attribute field (file header attributes), determines whether the data structure is associated with source code (file attribute associated with alternate code). Determines whether an attribute field in the data structure is associated with an attribute in the source code (time stamp or checksum information). If the source code associated with the attribute does not exist, the code is generated with associated attributes, but does not disclose determining a second attribute in the source code is not in the data structure and then removing the attribute from the source code. However, Timbol discloses removal of a property from source code (column 13, lines 11 – 13). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the teaching regarding the association of attributes in a data structure with source code in Hicks with the

removal of a property or attribute from source code as taught by Timbol because the combination provides a mechanism to modify the code as taught by Timbol (column 12, line 56) to reflect the data structure describing the code.

In regard to claim 4, incorporating the rejection of claim 3:

> *"...wherein the step of removing the second attribute from the source code comprises the step of removing a method associated with the second attribute from the source code."*

Hicks discloses the identification of a data structure with an attribute field (file header attributes), determines whether the data structure is associated with source code (file attribute associated with alternate code). Determines whether an attribute field in the data structure is associated with an attribute in the source code (time stamp or checksum information). If the source code associated with the attribute does not exist, the code is generated with associated attributes, but does not disclose determining a second attribute in the source code is not in the data structure and then removing the attribute from the source code. However, Timbol discloses removal of a method associated with a property from source code (column 13, lines 11 – 13). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the teaching regarding the association of attributes in a data structure with source code in Hicks with the removal of a method associated with a property or attribute from source code as taught by Timbol because the combination provides a mechanism to modify the code as taught by Timbol (column 12, line 56) to reflect the data structure describing the code.

In regard to claim 22, incorporating the rejection of claim 18:

> *"...wherein the step of removing the first attribute from the source code comprises the step of removing a method associated with the first attribute from the source code."*

Rejected for the same reasons put forth in the rejection of claim 4.

In regard to claim 24:

> *"receiving an indication that the data structure has been modified; and*
>
> *automatically reflecting the modification in the source code so as to avoid completely regenerating the source code."*

Hicks discloses updating of code if an item in the data structure is changed (column 8, lines 1 – 5).

In regard to claim 25, incorporating the rejection of claim 24:

> *"...wherein the step of automatically reflecting the modification comprises the steps of:*
>
> *determining whether a first attribute in the source code is associated with a first attribute field in the data structure; and*
>
> *when it is determined that a first attribute in the source code is not associated with a first attribute field in the data structure, removing the first attribute from the source code."*

Rejected for the same reason put forth in the rejection of claim 3.

In regard to claim 26, incorporating the rejection of claim 25:

> *"...wherein the step of removing the first attribute from the source code comprises the step of removing a first method associated with the first attribute in the source code."*

Rejected for the same reasons put forth in the rejection of claim 4.

In regard to claim 27, incorporating the rejection of claim 25:

> *"...wherein the step of automatically reflecting the modification further comprises the steps of:*
>
> > *determining whether a second attribute field in the data structure is associated with a second attribute in the source code; and*
> >
> > *when it is determined that a second attribute field in the data structure is not associated with a second attribute in the source code, generating the second attribute in the source code from the second attribute field and associating the second attribute with the second attribute field."*

Rejected for the same corresponding reasons put forth in the rejection of claim 18.

In regard to claim 28, incorporating the rejection of claim 27:

> *"...wherein the step of generating the second attribute in the source code comprises the step of generating a second method in the source code to access the second attribute field of the data structure."*

Rejected for the same reasons put forth in the rejection of claim 23.

In regard to claim 29, incorporating the rejection of claim 24:

> *"...further comprising the step of displaying a graphical representation of the source code."*

Rejected for the same reasons put forth in the rejection of claim 5.

In regard to claim 30, incorporating the rejection of claim 29:

> *"...further comprising the step of modifying the graphical representation of the source code to reflect the modification."*

Rejected for the same reasons put forth in the rejection of claim 6.

In regard to claim 31, incorporating the rejection of claim 24:

> *"...wherein the source code comprises a class."*

Rejected for the same reasons put forth in the rejection of claim 14.

In regard to claim 32, incorporating the rejection of claim 24:

"...wherein the source code comprises a distributed computing component."

Rejected for the same reasons put forth in the rejection of claim 15.

In regard to claim 33, incorporating the rejection of claim 32:

"...wherein the distributed computing component is an Enterprise JavaBean™."

Rejected for the same reasons put forth in the rejection of claim 16.

In regard to claim 34:

"determining whether the source code is associated with a data structure;

when it is determined that the source code is associated with the data structure, determining whether a first attribute in the source code is associated with a first attribute field of the data structure;

when it is determined that the first attribute in the source code is not associated with the first attribute field in the data structure, generating the first attribute field in the data structure;

determining whether a second attribute field in the data structure is associated with a second attribute in the source code; and

when it is determined that the second attribute field is not associated with the second attribute in the source code, removing the second attribute field from the data structure."

Hicks discloses the identification of a data structure with an attribute field (file header attributes), determines whether the data structure is associated with source code (file attribute associated with alternate code). If the data structure does not exist it is added or modified

(column 11, lines 13 – 16). Determines whether an attribute field in the data structure is associated with an attribute in the source code (time stamp or checksum information). If the source code associated with the attribute does not exist, the code is generated with associated attributes, but does not disclose determining a second attribute in the source code is not in the data structure and then removing the attribute from the source code. However, Timbol discloses removal of a property from source code (column 13, lines 11 – 13). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the teaching regarding the association of attributes in a data structure with source code in Hicks with the removal of a property or attribute from source code as taught by Timbol because the combination provides a mechanism to modify the code as taught by Timbol (column 12, line 56) to reflect the data structure describing the code.

In regard to claim 35, incorporating the rejection of claim 34:

> *"...further comprising the step of when it is determined that the source code is not associated with the data structure, generating the data structure from the source code."*
>
> See Hicks column 11, lines 13 – 16.

In regard to claim 36, incorporating the rejection of claim 34:

> *"...wherein the step of determining whether the source code is associated with the data structure comprises the step of searching a comment in the source code for an identification of the data structure."*
>
> Rejected for the same corresponding reason put forth in the rejection of claim 8.

In regard to claim 37, incorporating the rejection of claim 34:

> *"... wherein the step of determining whether the source code is associated with the data structure comprises the step of comparing a name for the source code with an identification of the data structure."*

Rejected for the same corresponding reason put forth in the rejection of claim 9.

In regard to claim 38, incorporating the rejection of claim 34:

"...further comprising the steps of:

retrieving access information for a database that stores the data structure; and

retrieving a portion of the data structure from the database using the access information."

Rejected for the same corresponding reason put forth in the rejection of claim 10.

In regard to claim 39, incorporating the rejection of claim 38:

"...wherein the step of retrieving the access information comprises the step of retrieving an identification of the data structure and the access information from a configuration file."

Rejected for the same corresponding reason put forth in the rejection of claim 11.

In regard to claim 40, incorporating the rejection of claim 38:

"...wherein the step of retrieving the access information comprises the step of retrieving an identification of the data structure and the access information from a comment of the source code."

Rejected for the same corresponding reason put forth in the rejection of claim 12.

In regard to claim 41, incorporating the rejection of claim 38:

"...wherein the portion of the data structure comprises the first attribute field in the data structure."

Rejected for the same corresponding reason put forth in the rejection of claim 13.

In regard to claim 42, incorporating the rejection of claim 34:

"...wherein the source code comprises a class."

Rejected for the same corresponding reason put forth in the rejection of claim 14.

In regard to claim 43, incorporating the rejection of claim 34:

"...wherein the source code comprises a distributed computing component."

Rejected for the same corresponding reason put forth in the rejection of claim 15.

In regard to claim 44, incorporating the rejection of claim 34:

"...wherein the first attribute field in the data structure is related to a method in the source code."

Rejected for the same corresponding reason put forth in the rejection of claim 17.

In regard to claim 45:

"receiving an indication to update a data structure related to the source code;

determining whether a first attribute field of the data structure is associated with a first attribute in the source code;

when it is determined that the first attribute field of the data structure is not associated with the first attribute of the source code, removing the first attribute field from the data structure;

determining whether a second attribute in the source code is associated with a second attribute field in the data structure; and

when it is determined that the second attribute is not associated with the second attribute field in the data structure, adding the second attribute field to the data structure."

Hicks discloses the identification of a data structure with an attribute field (file header attributes), determines whether the data structure is associated with source code (file attribute

associated with alternate code). If the data structure does not exist it ad added (column 11, lines 13 – 16) Determines whether an attribute field in the data structure is associated with an attribute in the source code (time stamp or checksum information). If the source code associated with the attribute does not exist, the code is generated with associated attributes, but does not disclose determining a second attribute in the source code is not in the data structure and then removing the attribute from the source code. However, Timbol discloses removal of a property from source code (column 13, lines 11 – 13). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the teaching regarding the association of attributes in a data structure with source code in Hicks with the removal of a property or attribute from source code as taught by Timbol because the combination provides a mechanism to modify the code as taught by Timbol (column 12, line 56) to reflect the data structure describing the code.

In regard to claim 46, incorporating the rejection of claim 45:

> *"...further comprising the steps of:*
>
> *retrieving access information for a database that stores the data structure; and*
>
> *retrieving a portion of the data structure from the database using the access information."*
>
> Rejected for the same corresponding reason put forth in the rejection of claim 10.

In regard to claim 47, incorporating the rejection of claim 46:

> *"...wherein the step of retrieving the access information comprises the step of retrieving an identification of the data structure and the access information from a configuration file."*

Rejected for the same corresponding reason put forth in the rejection of claim 11.

In regard to claim 48, incorporating the rejection of claim 46:

"...wherein the step of retrieving the access information comprises the step of retrieving an identification of the data structure and the access information from a comment of the source code."

Rejected for the same corresponding reason put forth in the rejection of claim 12.

In regard to claim 49, incorporating the rejection of claim 46:

"...wherein the portion of the data structure comprises the first attribute field in the data structure."

Rejected for the same corresponding reason put forth in the rejection of claim 13.

In regard to claim 50, incorporating the rejection of claim 45:

"...wherein the source code comprises a class."

Rejected for the same corresponding reason put forth in the rejection of claim 14.

In regard to claim 51, incorporating the rejection of claim 45:

"...wherein the source code comprises a distributed computing component."

Rejected for the same corresponding reason put forth in the rejection of claim 15.

In regard to claim 52:

"receiving an indication that the source code has been modified; and

automatically reflecting the modification in the data structure so as to avoid completely regenerating the data structure."

Hicks discloses the identification of a data structure with an attribute field (file header attributes), determines whether the data structure is associated with source code (file attribute associated with alternate code). If the data structure does not exist it is added or modified (column 11, lines 13 – 16). Determines whether an attribute field in the data structure is associated with an attribute in the source code (time stamp or checksum information). If the source code associated with the attribute does not exist, the code is generated with associated attributes, but does not disclose determining a second attribute in the source code is not in the data structure and then removing the attribute from the source code. However, Timbol discloses removal of a property from source code (column 13, lines 11 – 13). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the teaching regarding the association of attributes in a data structure with source code in Hicks with the removal of a property or attribute from source code as taught by Timbol because the combination provides a mechanism to modify the code as taught by Timbol (column 12, line 56) to reflect the data structure describing the code.

In regard to claim 53, incorporating the rejection of claim 52:

> *"...wherein the step of automatically reflecting the modification comprises the steps of:*
>
> *determining whether a first attribute in the source code is associated with a first attribute field of the data structure; and*
>
> *when it is determined that a first attribute is not associated with a first attribute field in the data structure, generating the first attribute field in the data structure."*

Hicks discloses the identification of a data structure with an attribute field (file header attributes), determines whether the data structure is associated with source code (file attribute associated with alternate code). Determines whether an attribute field in the data structure is

associated with an attribute in the source code (time stamp or checksum information). If the source code associated with the attribute does not exist, the code is generated with associated attributes (column 11, lines 6 – 21; column 10, lines 1 – 10; e.g., Figure 3).

In regard to claim 54, incorporating the rejection of claim 52:

> "...further comprising the steps of:
>
> determining whether a second attribute field in the data structure is associated with a second attribute in the source code; and
>
> when it is determined that a second attribute field is not associated with a second attribute in the source code, removing the second attribute field from the data structure."

Hicks discloses the identification of a data structure with an attribute field (file header attributes), determines whether the data structure is associated with source code (file attribute associated with alternate code). Determines whether an attribute field in the data structure is associated with an attribute in the source code (time stamp or checksum information). If the source code associated with the attribute does not exist, the code is generated with associated attributes, but does not disclose determining a second attribute in the source code is not in the data structure and then removing the attribute from the source code. However, Timbol discloses removal of a property from source code (column 13, lines 11 – 13). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the teaching regarding the association of attributes in a data structure with source code in Hicks with the removal of a property or attribute from source code as taught by Timbol because the combination provides a mechanism to modify the code as taught by Timbol (column 12, line 56) to reflect the data structure describing the code.

In regard to claim 55, incorporating the rejection of claim 52:

"...further comprising the steps of:

retrieving access information for a database that stores the data structure; and

retrieving a portion of the data structure from the database using the access information."

Rejected for the same reasons put forth in the rejection of claim 10.

In regard to claim 56, incorporating the rejection of claim 55:

"...wherein the step of retrieving the access information comprises the step of retrieving an identification of the data structure and the access information from a configuration file."

Rejected for the same reasons put forth in the rejection of claim 11.

In regard to claim 57, incorporating the rejection of claim 55:

"...wherein the step of retrieving the access information comprises the step of retrieving an identification of the data structure and the access information from a comment of the source code."

Rejected for the same reasons put forth in the rejection of claim 12.

In regard to claim 58, incorporating the rejection of claim 55:

"...wherein the portion of the data structure comprises the first attribute field in the data structure."

Rejected for the same reasons put forth in the rejection of claim 13.

In regard to claim 59, incorporating the rejection of claim 52:

"...wherein the source code comprises a class."

Rejected for the same reasons put forth in the rejection of claim 14.

In regard to claim 60, incorporating the rejection of claim 52:

"...wherein the source code comprises a distributed computing component."

Rejected for the same reasons put forth in the rejection of claim 15.

In regard to claim 61, incorporating the rejection of claim 52:

"...wherein the first attribute field in the data structure is related to a method in the source code."

Rejected for the same reasons put forth in the rejection of claim 17.

Claims 64 and 65 (computer readable medium) are rejected for the same corresponding reasons put forth in the rejection of claims 3 and 4 (the corresponding methods).

Claim 83 (computer readable medium) is rejected for the same corresponding reasons put forth in the rejection of claim 22 (the corresponding methods).

Claims 85 - 94 (computer readable medium) are rejected for the same corresponding reasons put forth in the rejection of claims 24 – 33 (the corresponding methods).

Claims 95 - 105 (computer readable medium) are rejected for the same corresponding reasons put forth in the rejection of claim 34 – 44 (the corresponding methods).

Claims 106 - 112 (computer readable medium) are rejected for the same corresponding reasons put forth in the rejection of claim 45 – 51 (the corresponding methods).

Claims 113 - 122 (computer readable medium) are rejected for the same corresponding reasons put forth in the rejection of claim 52 – 61 (the corresponding methods).

Claims 129 - 135 (a data processing system) are rejected for the same corresponding reasons put forth in the rejection of claims 45 – 51 (the corresponding methods).

Claim 136 (a system) is rejected for the same corresponding reasons put forth in the rejection of claim 24 (the corresponding method).

6. Claims 5 – 7; and 66 – 68 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hicks, U.S. Patent 6,654,954 in view of Timbol, U.S. Patent 6,237,135, and further in view of Mansurov et al., U.S. Patent 6,346,945 (hereinafter referred to as Mansurov).

In regard to claim 5, incorporating the rejection of claim 3:

> *"...further comprising the step of displaying a graphical representation of the source code."*

Neither Hicks nor Timbol discloses a graphical representation of code. However, graphically representing source code is well known in the art as exemplified by Mansurov (column 3, lines 39 – 54). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the teaching regarding the association of attributes in a

data structure with source code in Hicks with the removal of a method associated with a property or attribute from source code as taught by Timbol because the combination provides a mechanism to modify the code as taught by Timbol to reflect the data structure describing the code and further modified by the graphical representation of the code as taught by Mansurov because a software developer would be motivated to visually lay out the code so that a clear understanding of relationships between procedures and functions as well as relationships between classes (see Mansurov, column 1, lines 16 – 24).

In regard to claim 6, incorporating the rejection of claim 5:

> *"...further comprising the step of modifying the graphical representation of the source code to reflect the generation of the new attribute.*

Neither Hicks nor Timbol discloses modifying a graphical representation of the source code to reflect generation of the new attribute. However, Mansurov discloses representing source code with a graphical representation, which includes attribute computation (Abstract; column 3, lines 39 – 54). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the teaching regarding the association of attributes in a data structure with source code in Hicks with the removal of a method associated with a property or attribute from source code as taught by Timbol because the combination provides a mechanism to modify the code as taught by Timbol to reflect the data structure describing the code and further modified by the graphical representation of the code and computation of attributes as taught by Mansurov because a software developer would be motivated to visually lay out the code so that a clear understanding of relationships between procedures and functions

as well as relationships between classes as the code is changed (see Mansurov, column 1, lines 16 – 24).

In regard to claim 7, incorporating the rejection of claim 5:

> *"...further comprising the step of modifying the graphical representation of the source code to reflect the removal of the second attribute."*

Neither Hicks nor Timbol discloses modifying a graphical representation of the source code to reflect generation of the new attribute. However, Mansurov discloses representing source code with a graphical representation, which includes attribute computation (Abstract; column 3, lines 39 – 54). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the teaching regarding the association of attributes in a data structure with source code in Hicks with the removal of a method associated with a property or attribute from source code as taught by Timbol because the combination provides a mechanism to modify the code as taught by Timbol to reflect the data structure describing the code and further modified by the graphical representation of the code and computation of attributes as taught by Mansurov because a software developer would be motivated to visually lay out the code so that a clear understanding of relationships between procedures and functions as well as relationships between classes as the code is changed (see Mansurov, column 1, lines 16 – 24). The graphical representation can be run to reflect any code changes.

Claims 66 - 68 (computer readable medium) are rejected for the same corresponding reasons put forth in the rejection of claims 5 – 7 (the corresponding methods).

7. Claims 8, 12; 19; 69, 73; and 80 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hicks, U.S. Patent 6,654,954 in view of Foster, U.S. Publication No. US 2002/0078071.

In regard to claim 8, incorporating the rejection of claim 1:

> *"...wherein the step of determining whether the data structure is associated with the source code comprises the step of searching a comment in the source code for the identification of the data structure."*

Hicks discloses the identification of a data structure with an attribute field (file header attributes), determines whether the data structure is associated with source code (file attribute associated with alternate code). Determines whether an attribute field in the data structure is associated with an attribute in the source code (time stamp or checksum information). If the source code associated with the attribute does not exist, the code is generated with associated attributes, but does not disclose searching a comment in the source code for the identification of a data structure associated with the source code. However, Foster discloses searching the source code comments to identify an attribute (paragraph [0029]). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the teaching regarding the association of attributes in a data structure with source code in Hicks with searching the source code for attribute information as taught by Foster because the combination provides a mechanism to search the code for the data structure information, which would eliminate a requirement for a separate list as taught by Foster at paragraph [0029].

In regard to claim 12, incorporating the rejection of claim 10:

> *"...wherein the step of retrieving the access information comprises the step of retrieving the identification of the data structure and the access information from a comment of the source code."*

Rejected for the same reasons put forth in the rejection of claim 8.

In regard to claim 19, incorporating the rejection of claim 18:

> *"... further comprising the step of retrieving an identification of the data structure and access information for the secondary storage from a comment in the source code."*

Rejected for the same reason put forth in the rejection of claim 8.

Claims 69 and 73 (computer readable medium) are rejected for the same corresponding reasons put forth in the rejection of claims 8 and 12 (the corresponding methods).

Claim 80 (computer readable medium) is rejected for the same corresponding reasons put forth in the rejection of claim 19 (the corresponding methods).

Conclusion

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lawrence Shrader whose telephone number is (703) 305-8046. The examiner can normally be reached on M-F 08:00-16:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703) 305-9662. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Lawrence Shrader
Examiner
Art Unit 2124

28 May 2004

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100